A Novel Approach for Internet Traffic Classification based on Multi-Objective Evolutionary Fuzzy Classifiers

Pietro Ducange*, Giuseppe Mannarà†, Francesco Marcelloni†, Riccardo Pecori* and Massimo Vecchio*
*SMARTEST Research Centre, eCampus University, Via Isimbardi 10, 22060 Novedrate (Como), Italy
Email:{name.surname}@uniecampus.it
† Department of Information Engineering, University of Pisa, Largo Lucio Lazzarino 1, 56122 Pisa, Italy
Email:{name.surname}@iet.unipi.it

***Abstract*—Internet traffic classification has moved in the last years from traditional port and payload–based approaches towards methods employing statistical measurements and machine learning techniques. Despite the success achieved by these techniques, they are not able to explain the relation between the features, which describe the traffic flow, and the corresponding traffic classes. This relation can be extremely useful to network managers for quickly handling possible network drawback. In this paper, we propose to tackle the traffic classification problem by using multi-objective evolutionary fuzzy classifiers (MOEFCs). MOEFCs are characterised by good trade-offs between accuracy and interpretability. We adopt two Internet traffic datasets extracted from two real-world networks. We discuss the results obtained both by applying a cross validation on each single dataset, and by using a dataset as training set and the other as test set. We show that, in both cases, MOEFCs can achieve satisfactory accuracy in the face of low complexity and, therefore, high interpretability.**

I. Introduction

Nowadays, the classification of the Internet traffic is, together with its management, one of the most important tasks carried out by Internet Service Providers. Although historically this task was mainly performed for security purposes, as it supports the detection and the identification of intrusions and malicious behaviors, recently it is gaining new momentum. Indeed, since it is able to reliably characterize the Internet traffic and its workloads, it is currently one of the most cost-effective ways to perform traffic engineering and to take decision on policing, traffic shaping, billing, dynamic Quality of Service and so on. In general, network traffic flows can be classified according to different granularity levels. Nevertheless, especially in recent years, one of the most popular classification challenges in this field regards the task of discovering the specific application-layer protocol that generated a certain network flow (simply *"flow"*, in the following) [1].

In the literature, two main approaches have been used for the Internet traffic classification (ITC), namely port-based and payload-based approaches. While port-based ITC techniques rely on the observation of server port numbers (information carried in the transport-layer header) to map flows to application-layer protocols following the assignment made by the Internet Assigned Numbers Authority[1], payload-based ITC techniques imply deep packet inspections (of both header and/or payload) to look for signatures and fingerprints that are typical of certain application-layer protocols [2]. Unfortunately, in the last decade, there has been a steep growth of computer and network applications, which use non-standard ports and encrypted traffic. This new trend has deeply affected the classification performances of any port-based and payload-based ITC technique. With the aim of coping with this growth, two main categories of approaches are emerging in the ITC scenario: statistical methods and behavioral classifiers. While the latter regard the observation of the whole traffic received by a host, or an endpoint, in the network, the former are based onto concepts of statistics and information theory to characterize flows, sessions or even isolated packets themselves [3].

Statistical techniques take advantage of different machine learning and data mining techniques to perform ITC; moreover, depending on whether the available traffic traces for building the dataset are labeled or not they may be based on predictive or descriptive models [4]. In the first case, only previously known application-layer protocols may be identified, while in the second one novel patterns and classes can be discovered [5], [6]. Analyzing the specialized literature [7], [8], [9], we highlight that the use of data mining and machine learning methods for ITC is quite popular, hence still representing a hot research topic.

The main drawback of the aforementioned approaches, including the ones based on data mining and machine learning, is that the generated models are *black boxes* characterized by a low *"interpretability"* level. Indeed, researchers and practitioners cannot extract any useful information regarding how the Internet traces have been classified. On the other hand, it could be very useful to extract, analyze and exploit the insights hidden in the classification models. For this reason, in this work, we propose a traffic classification approach based on multi-objective evolutionary fuzzy classifiers (MOEFCs) [10], [11]. Since MOEFCs are characterized by good trade-off between their accuracy and their interpretability level, these

[1]IANA, http://www.iana.org/

978-1-5090-6034-4/17/$31.00

models have been widely used for approaching classification problems. Indeed, MOEFCs deal with the design of fuzzy rule-based classifiers (FRBCs) by means of multi-objective evolutionary algorithms: during the evolutionary design process, both the accuracy and the interpretability level of the models are concurrently optimized. The final models are usually characterized by compact fuzzy rules, namely linguistic IF-THEN rules, which are able to describe the classification process in an interpretable way.

To evaluate the proposed approach, we use two real-world Internet traffic datasets. In particular, we first carry out a five-fold cross validation analysis on each dataset. Then, we perform a cross-network validation, that is, we use one dataset for training the classification model and the other for testing it. Results show that the classification models achieve accuracies up to 93 % and 74%, in the cross-validation and cross-network validation experiments, respectively. Moreover, the generated models are characterized by high interpretability levels.

The paper is organized as follows: in Sec. II we describe the flows and the features used to classify them; in Sec. III we characterize the traces of the analyzed networks. Sec. IV introduces the MOEFCs and Sec. V discusses the achieved results, giving some comments on the extracted knowledge. Finally, Sec. VI draws some conclusions and gives possible future research directions.

II. Traffic Flows Description

Generally, when dealing with data mining and machine learning approaches for traffic classification, authors consider scenarios where a bi-directional network flow is composed by a distinguishable stream of ordered packets, exchanged between two endpoints, identified by the following common quintuple: {*source IP address*, *destination IP address*, *source port*, *destination port*, *transport protocol*}, where source and destination ports and addresses may be pairwise interchangeable. Indeed, traffic classification platforms may take advantage of network sniffers and analyzing tools, such as Wireshark[2] or tcpdump[3]. These tools allow catching traces of Internet traffic, flowing inside private or public networks: traces are composed of various packets belonging to different sessions or flows.

In the literature, the statistical traffic classification approaches usually perform their tasks at two different layers: at a fine-grained level the procedure is able to identify the particular *application protocol* that generated a certain flow, while at a coarse-grained level it is able to identify only a group of protocols (*e.g.*, bulk transfer, mailing, web browsing, etc). Whatever the granularity of the analysis is, the classification algorithms usually consider transfer-based, time-based and signaling-based features of packets to characterize a flow (but the same holds for sessions) [12].

As discussed in [5], the description of flows in terms of numerical features may be carried out at different levels of abstraction. On the one hand, there exist methods that look at packets and at their inherent and simplest properties, such as their size expressed in Bytes, their relative temporal distance with respect to the beginning of the flow, their relative position within the flow. In this case, it is reasonable to consider packets and flows as bi-dimensional random variables and stochastic processes, respectively. On the other hand, there are methods that rely on some aggregated statistical features characterizing the uni-directional components of a flow (*sub-flow*, in the following), such as maximum, minimum, mean and standard deviation of certain quantities of the sub-flow (*e.g.*, the packet size, the inter-arrival time, etc.), together with certain other quantities of the whole flow (*e.g.*, the total volume in Bytes, the overall duration in milliseconds, etc.). In this paper, we focus on the latter approach, namely the one encompassing aggregated statistical features of the flows (and their components). With respect to the granularity, the approach we are going to describe belongs to the fine-grained ITC family, since it aims to classify network flows as specific application-layer protocols rather than classes of these protocols.

Specifically, to perform our quantitative analysis, we start from traffic flows and extract a set of statistical features, as summarized in Table I: these features have been already successfully adopted in the traffic classification schemes discussed in [5], [6]. Table I lists, for each feature, a brief description, the unit of measurement (U/M) of the feature (where ms and B stand for milli-seconds and Bytes, respectively) and the label used in the following to refer to it. For the sake of the readability, we use prefixes $f\_$ and $r\_$ to distinguish between the directions of the sub-flows (*e.g.*., "forward" and "reverse"), while subscripts are used to specify the type of aggregation (m, M, μ and σ for minimum, maximum, mean and standard deviation, respectively). Hence, except for the duration, all the features used in this paper are based on the sub-flow; the total number of features is 21.

TABLE I
The list of the 21 features used to characterize a flow.

Description of the feature	U/M	Features	
		forward	reverse
Flow duration	ms	Δ	
Number of transferred packets	-	f_N	r_N
Transferred volume	B	f_V	r_V
Minimum packet size	B	f_S_m	r_S_m
Maximum packet size	B	f_S_M	r_S_M
Average packet size	B	f_S_μ	r_S_μ
Standard deviation of packet size	B	f_S_σ	r_S_σ
Minimum inter-packet time	ms	f_T_m	r_T_m
Maximum inter-packet time	ms	f_T_M	r_T_M
Average inter-packet time	ms	f_T_μ	r_T_μ
Standard deviation of inter-packet time	ms	f_T_σ	r_T_σ

III. The Analyzed Internet Traffic Datasets

In our experiments, we adopted two different packet traces, namely KEIO and WIDE traces, which come from two different network locations and temporal periods. Both traces are

[2]http://www.wireshark.org/
[3]http://www.tcpdump.org/

provided by the public traffic data repository maintained by the MAWI Working Group of the WIDE Project[4].

The KEIO trace refers to 1 Gbps Ethernet link traffic of a Japanese campus captured in August 2006, while the WIDE trace encompasses 2008 trans-Pacific backbone traces from USA and Japan and vice versa. The characterization of both traces is provided in Table II: shortly, KEIO presents 6 application-layer protocol classes, namely BitTorrent (BT), HTTP, IMAP, POP3, RAZOR and SMTP, while WIDE includes 8 application-layer protocol classes, namely BT, FTP, HTTP, IMAP, POP3, RAZOR, SMTP and SSH. Similar to the work in [5], where both the KEIO and WIDE traces have been analyzed, we randomly sample 500 flows for each application-layer protocol.

An important aspect, when assessing the reliability of a generic traffic classifier, is the provenance of the so-called ground truth information: indeed, it is necessary to know the actual application-layer protocol associated with a specific flow for assessing the correctness of the estimated protocol. In the past, when dealing with application-layer protocols using standard port numbers, such numbers could determine directly the actual protocol. Nowadays, deep packet inspection (DPI) tools are generally used to build up reliable ground truth information. In this paper, we adopt the same ground truth procedure used in [5] and based on DPI. Due to space limitation, we cannot describe the procedure: the interested reader can refer to [5] for details.

TABLE II
CHARACTERISTICS OF THE TWO INTERNET TRAFFIC DATASETS.

Network	Link type	# application-layer protocols	application-layer protocols (aka classes)
WIDE	backbone	8	BT, FTP, HTTP, IMAP, POP3, RAZOR, SMTP, SSH
KEIO	edge	6	BT, HTTP, IMAP, POP3, RAZOR, SMTP

IV. MULTI-OBJECTIVE EVOLUTIONARY FUZZY CLASSIFIERS

In this work, we approach the traffic classification problem adopting Multi-objective Evolutionary Fuzzy Classifiers (MOEFCs) [10], [11]. MOEFCs represent the hybridization of Multi-Objective Evolutionary Algorithms (MOEAs) and Fuzzy Rule-based Classifiers (FRBCs). MOEAs have been widely and successfully used in the last years for designing the architecture of FRBCs. During the evolutionary optimization process, in general two main objectives are concurrently optimized, namely the accuracy and the interpretability, which are in conflict with each other.

An FBRC basically includes a rule base (RB), a database (DB) containing the definition of the fuzzy sets used in the RB, and an inference engine. RB and DB comprise the knowledge base of the rule-based system.

Let $X = \{X_1, \ldots, X_F\}$ be the set of input variables and X_{F+1} be the output variable of the classifier. Let U_f, with $f = 1, ..., F$, be the universe of the f^{th} input variable X_f. Let $P_f = \{A_{f,1}, \ldots, A_{f,j}, \ldots, A_{f,T_f}\}$ be a partition of variable X_f consisting of T_f fuzzy sets. The output variable X_{F+1} is a categorical variable assuming values in the set Γ of K possible classes $\Gamma = \{C_1, \ldots, C_K\}$. Let $\{(\mathbf{x}_1, x_{F+1,1}), \ldots, (\mathbf{x}_N, x_{F+1,N})\}$ be a training set composed of N input-output pairs, with $\mathbf{x}_t = [x_{t,1} \ldots, x_{t,F}] \in \Re^F$, $t = 1, \ldots, N$ and $x_{F+1,t} \in \Gamma$.

With the aim of determining the class of a given input vector, we adopt an RB composed of M rules expressed as:

$$R_m : \textbf{IF } X_1 \textbf{ is } A_{1,j_{m,1}} \textbf{ AND} \ldots \textbf{AND } X_F \textbf{ is } A_{F,j_{m,F}} \textbf{ THEN } X_{F+1} \textbf{ is } C_{j_m} with\ RW_m \quad (1)$$

where C_{j_m} is the class label associated with the m^{th} rule, and RW_m is the rule weight, *i.e.*, a certainty degree of the classification in the class C_{j_m} for a pattern belonging to the subspace delimited by the antecedent of rule R_m.

Usually, a purposely-defined fuzzy set $A_{f,0}$ ($f = 1, \ldots, F$) is considered for all the F input variables. This fuzzy set, which represents the "don't care" condition, is defined by a membership function equal to 1 on the overall universe. The term $A_{f,0}$ allows generating rules that contain only a subset of the input variables.

A specific *reasoning method* uses the information from the RB to determine the class label for a given input pattern. We adopt the *maximum matching* as reasoning method (see [13] for details).

As regards the DB, we adopt triangular fuzzy sets: each fuzzy set $A_{f,j}$ is identified by the tuples $(a_{f,j}, b_{f,j}, c_{f,j})$, where $a_{f,j}$ and $c_{f,j}$ correspond to the left and right extremes of the support, and $b_{f,j}$ to the core. In particular, in the experiments, we use strong fuzzy partitions, where $a_{f,1} = b_{f,1}, b_{f,T_f} = c_{f,Tf}$ and, for $j = 2, ..., T_f - 1$, $b_{f,j} = c_{f,j-1}$ and $b_{f,j} = a_{f,j+1}$.

In order to concurrently design the RB and tune the parameters of the fuzzy sets, we adopt the PAES-RCS algorithm introduced in [13]. The multi-objective evolutionary learning scheme is based on the (2+2)M-PAES, which is an MOEA successfully employed in the last years in the context of MOEFSs. We concurrently optimize two objectives: the first objective considers the intepretability of the RB, calculated as the total rule length (TRL), that is, the number of propositions used in the antecedents of the rules contained in the RB; the second objective takes into account the accuracy, assessed in terms of classification rate.

In the learning scheme, we first generate an initial RB and then select, during the evolutionary process, the most relevant rules and conditions in the rules. Moreover, we concurrently tune the parameters of the fuzzy sets by using a mapping strategy based on a *piecewise linear transformation* [14]. Once defined an initial strong fuzzy partition for each input variable, we extract the initial RB from a decision tree: in particular, in this work, we use a recent algorithm, discussed in [15], for generating multi-way fuzzy decision trees. One rule is created for each path from the root to a leaf node.

[4]MAWI Working Group Traffic Archive. URL: http://mawi.wide.ad.jp/mawi/

In PAES-RCS each solution is codified by a chromosome C composed of two parts (C_R, C_T), which define, respectively, the RB and the positions of the representatives of the fuzzy sets, namely the cores, in the transformed space.

Let J_{DT} and M_{DT} be the initial RB generated by the decision tree and the number of rules of this RB, respectively. In order to generate compact and interpretable RBs, we allow that the RB of a solution contains at most M_{max} rules. The C_R part, which codifies the RB, is a vector of M_{max} pairs $\mathbf{p}_m = (k_m, \mathbf{v}_m)$, where $k_m \in [0, M_{DT}]$ identifies the selected rule of J_{DT} and $\mathbf{v}_m = [v_{m,1}, \ldots, v_{m,F}]$ is a binary vector which indicates, for each variable X_f, if the condition is present or not. In particular, if $k_m = 0$ the m^{th} rule is not included in the RB. Thus, we can generate RBs with a lower number of rules than M_{max}. Further if $v_{m,f} = 0$ the f^{th} condition of the m^{th} rule can be replaced by a *"don't care"* condition.

C_T is a vector containing F vectors of $T_{max} - 2$ real numbers: the f^{th} vector $[b_{f,2}, \ldots, b_{f,T_{max}-1}]$ determines the positions of the fuzzy set representatives in the specific variable X_f.

In order to generate the offspring populations, we exploit both crossover and mutation. We apply separately the one-point crossover to C_R and the BLX-α-crossover, with α = 0.5, to C_T. As regards the mutation, we apply two distinct operators for C_R and an operator for C_T. More details regarding the mating operators and the steps of PAES-RCS can be found in [13], [14].

V. Experimental analysis

In this section, we discuss the results achieved by the adopted MOEFCs on ITC problems. In particular, we first show the results obtained by considering the two datasets discussed in Section III separately. For each dataset, we performed a five-fold cross-validation and executed three trials for each fold with different seeds for the random function generator (15 trials in total). Then, we build MOEFCs by using one dataset, extracted from a network, and then test them on the other dataset extracted from the other network. In this way, we aim to evaluate the generalization capability of the MOEFCs on traffic data extracted from a network different from the one used for the training phase. We denote this experiment as *cross-network evaluation*.

Table III shows the parameters of PAES-RCS used in the experiments.

TABLE III
Values of the parameters used in the experiments.

Parameter	short name	value
Total number of fitness evaluations	N_{val}	50000
(2+2)M-PAES archive size	AS	64
Maximum number of rules in a RB	M_{max}	50
Probability of applying crossover operator to C_R	P_{C_R}	0.1
Probability of applying crossover operator to C_T	P_{C_T}	0.5
Probability of applying first mutation operator to C_R	P_{MRB_1}	0.1
Probability of applying second mutation operator to C_R	P_{MRB_2}	0.7
Probability of applying mutation operator to C_T	P_{M_T}	0.2
Number of fuzzy sets for each linguistic variable	T_f	5

Since several solutions can lie on the Pareto front approximations, typically only some representative solutions are considered in the comparison. Like the analysis carried out in [13], for each fold and each trial, the Pareto front approximations of each algorithm are computed and the solutions in each approximation are sorted according to decreasing accuracies on the training set. Then, for each approximation, we select the first (the most accurate), the median and the last (the least accurate) solutions. We denote these solutions as FIRST, MEDIAN and LAST, respectively.

Table IV summarizes the average results achieved by PAES-RCS on the WIDE and KEIO networks. Specifically, it shows the average values of the accuracy on training, Acc_{TR}, and test, Acc_{TST}, sets, respectively, TRL, number of rules ($\#R$), number of features ($\#F$) of the FIRST, MEDIAN and LAST solutions.

TABLE IV
Average results achieved by PAES-RCS on the WIDE and KEIO networks.

NET	SOLUTION	Acc_{TR}	Acc_{TST}	TRL	$\#R$	$\#F$
WIDE	FIRST	90.05%	88.36%	178.47	45.60	19.00
	MEDIAN	86.61%	85.44%	84.43	24.60	17.33
	LAST	53.90%	53.51%	26.53	11.67	11.13
KEIO	FIRST	93.53%	90.79%	160.80	43.13	18.00
	MEDIAN	91.02%	88.50%	73.20	22.47	15.73
	LAST	55.58%	54.87%	24.73	11.60	10.87

From Table IV, we can observe that the average FIRST solutions, though requiring a quite limited number of rules (*i.e.*, lower than 46 and 44 for WIDE and KEIO, respectively), is able to achieve satisfactory results (*i.e.*, more than 88% of the accuracy on the test set for the WIDE dataset and more than 90% for the KEIO one). The results achieved by the MEDIAN solutions are even more interesting: on average, using just less of half rules with respect to the FIRST counterparts (24.60 and 22.47 for WIDE and KEIO networks, respectively), a MEDIAN solution is able to reach accuracies on test sets which are only less than 4% worse than the FIRST counterpart (85.44% and 88.50% are the absolute MEDIAN accuracies, computed for the WIDE and KEIO networks, respectively). Even though the LAST solutions are characterized by a low complexity level, they achieve poor accuracies. Thus, we must discard them for traffic classification tasks.

We highlight that the shown results, in terms of classification accuracy, are better than the ones achieved by a state-of-the-art fuzzy classifier, namely FARC-HD [16], and slightly worse than the ones achieved by the C4.5 decision tree classifier [17]. Indeed, the FARC-HD achieves average accuracies on the test set equal to 79.60% and to 78.53% for WIDE and KEIO, respectively. On the other hand, the C4.5 achieves average accuracies on the test set equal to 93.75% and to 92.01% for WIDE and KEIO, respectively. We recall that the C4.5 is not a linguistic classifier and its interpretability level is much more lower than the one of an FRBC. Moreover, the average size of the generated trees, which more or less corresponds to the TRL, is equal to 233 and to 151 for WIDE and KEIO, respectively.

TABLE V

TRUE POSITIVE RATE (TPR) AND FALSE POSITIVE RATE (FPR) EXPRESSED IN PERCENTAGE, FOR THE BEST, MEDIAN AND LAST SOLUTIONS, ON THE TRAINING AND TEST SETS (TRAIN AND TEST) OF WIDE AND KEIO NETWORK TRACES.

CLASS			BT		FTP		HTTP		IMAP		POP3		RAZOR		SMTP		SSH	
NET	SET	SOL	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
WIDE	TRAIN	BEST	86.30	1.94	93.38	0.95	86.32	1.94	88.10	1.70	85.72	2.04	97.80	0.31	83.03	2.42	99.72	0.04
		MEDIAN	83.80	2.30	91.27	1.26	82.00	2.54	81.70	2.61	80.13	2.82	96.78	0.46	78.25	3.08	99.77	0.03
		LAST	74.48	3.84	74.28	3.80	54.37	6.17	35.07	8.64	45.17	7.61	47.08	7.06	34.63	8.66	66.12	4.27
	TEST	BEST	84.47	2.21	92.33	1.10	83.80	2.29	86.13	1.99	83.20	2.40	97.47	0.36	79.80	2.87	99.67	0.05
		MEDIAN	82.00	2.56	90.93	1.30	81.60	2.60	79.53	2.91	77.87	3.14	96.13	0.55	76.40	3.34	99.80	0.03
		LAST	74.27	3.82	73.40	3.92	55.07	6.05	32.87	8.90	45.40	7.58	47.60	6.99	33.53	8.80	65.93	4.30
KEIO	TRAIN	BEST	96.88	0.62	-	-	91.67	1.65	92.80	1.44	88.32	2.31	98.65	0.27	92.85	1.44	-	-
		MEDIAN	95.42	0.92	-	-	90.35	1.92	89.47	2.10	84.30	3.10	98.03	0.39	88.73	2.25	-	-
		LAST	61.75	7.64	-	-	55.35	8.14	45.75	9.76	53.98	8.62	67.45	5.86	49.22	9.53	-	-
	TEST	BEST	94.53	1.09	-	-	89.73	2.04	89.13	2.14	84.13	3.13	97.40	0.52	89.80	2.05	-	-
		MEDIAN	94.00	1.20	-	-	90.07	1.98	83.80	3.16	80.73	3.81	96.87	0.63	85.80	2.83	-	-
		LAST	64.33	7.22	-	-	55.33	8.13	42.27	10.54	51.27	9.09	68.13	5.71	47.87	9.70	-	-

To carry out an overall per class analysis of the performance achieved by the generated FRBCs, Table V shows, for each network, the average per class True Positive Rate (TPR) and False Positive Rate (FPR), associated with the FIRST, MEDIAN and LAST solutions. As regards the WIDE network, on the test set, FTP, RAZOR and SSH flows are better recognized than the others. Further, the TPRs on the test set are generally higher than 80% and 76% for FIRST and MEDIAN solutions, respectively. The FPRs span between 0.03% and 3.34% for the FIRST and MEDIAN solutions. As regards the KEIO network, on the test set, the highest TPRs and FPRs are achieved on BT and RAZOR flows, for both the FIRST and MEDIAN solutions. The remaining TPRs and FPRs span, respectively, between 80% and 90%, and 1.98% and 3.80%, for the FIRST and MEDIAN solutions.

To discuss in detail a specific FRBC generated by applying PAES-RCS, we focus on a single Pareto front approximation obtained in the first trial and first fold over the WIDE network and we analyze the MEDIAN solution. This solution is characterized by 24 rules and 17 features, with a TRL of 81, and achieves accuracy values of 88.91% and 88.38% on the training and test sets, respectively.

Fig. 1 shows a graphical version of the confusion matrix associated with the chosen solution when classifying the test set. By observing this graph, we can deduce that the MEDIAN solution was able to correctly classify 99% of the SSH flows; RAZOR, FTP and HTTP flows were also quite reliably classified (96%, 95% and 89% of correct classification, respectively); then the performances degrades towards 80%, when classifying the remaining application flows (83% for SMTP and POP3, 82% for BT and 80% for IMAP).

Fig. 2 summarizes how the rules are distributed among the classes, that is, the number of rules which have a specific class in the consequent. By analyzing the information contained in Figs. 1-2, we can draw interesting conclusions: the most impressive is that the selected solution actually employs only one rule to correctly classify the 99% of the SSH flows. Further, the solution uses 3 rules to correctly classify the 96% (resp. the 95%) of the RAZOR (resp. the FTP) flows. Also, the solution uses 4 rules to correctly classify the 80% of the IMAP flows.

Taking into account that the main goal of this preliminary work was to build (and rely on) more interpretable models for accurate traffic flows classification, due to space limit, we show only the rule used for classifying SSH in (2). We re-labeled the fuzzy sets as "very low", "low", "medium", "high" and " very high" (*VL*, *L*, *M*, *H*, *VH*, respectively), while maintaining the feature labels of Table I.

$$R_{12}: \textbf{IF } f\_S_m \textbf{ is } M \textbf{ AND } f\_S_M \textbf{ is } M \textbf{ AND } f\_S_\sigma \textbf{ is } H \textbf{ AND } f\_N \textbf{ is } H \textbf{ AND } r\_S_m \textbf{ is } M \textbf{ THEN } Class \textbf{ is } SSH \quad (2)$$

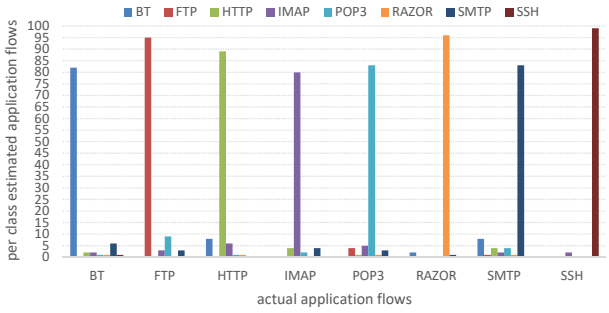

Fig. 1. A graphical view of the confusion matrix on the test set associated with the selected solution for the WIDE network.

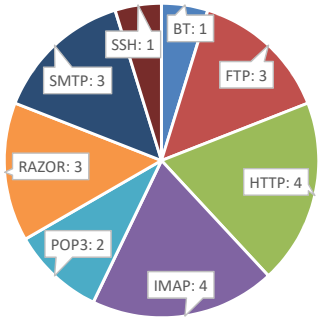

Fig. 2. Distribution of the rules among the classes for the selected MEDIAN solution.

Finally, in Table VI we show the results achieved when we use the dataset extracted from one network as training set, and the other dataset as test set. In the table, the label KEIO_TO_WIDE (WIDE_TO_KEIO) states that the classifiers were trained by using KEIO (resp. WIDE) and tested

TABLE VI
AVERAGE RESULTS ACHIEVED BY PAES-RCS FOR CROSS-NETWORK EVALUATION.

CLASS			BT		HTTP		IMAP		POP3		RAZOR		SMTP	
Experiment	**SOL**	Acc_{TST}	**TPR**	**FPR**	**TPR**	**FPR**	**TPR**	**FPR**	**TPR**	**FPR**	**TPR**	**FPR**	**TPR**	**FPR**
KEIO_TO_WIDE	BEST	74.09	79.93	4.10	87.73	2.53	83.13	3.66	48.67	9.56	86.80	2.53	58.27	7.92
	MEDIAN	72.99	70.80	5.75	84.27	3.19	83.73	3.62	49.13	9.50	91.00	1.77	59.00	7.81
WIDE_TO_KEIO	BEST	69.41	59.20	7.75	77.87	4.29	66.47	6.92	69.67	6.54	71.07	5.43	72.20	5.55
	MEDIAN	67.77	53.40	8.74	73.93	5.03	69.20	6.47	66.00	7.31	78.00	4.22	66.07	6.78

by using WIDE (resp. KEIO). In the experiments, in order to deal with the same number of classes, we removed the FTP and SSH flows from the WIDE network. For the sake of brevity, in the table we show only the per class TPRs and FPRs and the accuracies on the test set. The complexities of the models correspond more or less to the ones shown in Table IV. Moreover, as stated before, we do not show the LAST solutions.

Table VI shows that the overall accuracies on the test networks drastically decreases in comparison with the accuracies on the test set shown in Table IV. We must recall, as stated in Section III, that we are considering two networks that are very different. Thus, we might expect even worse accuracies associated with the test networks. Indeed, the achieved accuracies span between 67.77% and 74.09%. If we check the TPRs associated with the KEIO_TO_WIDE network, except for the POP3 and SMTP flows, they are generally higher than 80%. As regards the WIDE_TO_KEIO network, we observe that the performance deteriorates even more. On the other hand, we highlight that the TPRs, for the HTTP, the RAZOR and the SMTP flows are generally higher than 70%.

VI. CONCLUSION

In this paper, we have discussed a preliminary study on the application of multi-objective evolutionary fuzzy classifiers for approaching the Internet traffic classification problem. In particular, we have concentrated our efforts on the possibility of generating interpretable classification models, also characterized by a good accuracy level. Indeed, we have experimented a state-of-the-art algorithm, namely PAES-RCS, for generating sets of fuzzy rule-based systems for classifying the Internet traffic flows extracted from two real-world networks. We have first performed a cross validation, considering the data extracted from each network as an independent classification dataset. Then, we have carried out a cross-network validation, where one network has been used for training the classifiers and the other network has been adopted for evaluating the generalization capability of the trained models.

We have shown that the results achieved on the cross validation are promising. Indeed, the obtained fuzzy rule-based classifiers are characterized by very interesting trade-offs between their accuracy, expressed in terms of classification rate and also in terms of per class true positive and false positive rates, and their interpretability, expressed in terms of number of rules and total number of antecedents in each rules. As regards the cross-network evaluation, even though we have observed a substantial decrease of the accuracies, the generated classification models still maintain acceptable levels of classification rate, true positive rate and false positive rate.

Future works will cover a more detailed analysis of more recent application-layer protocols, considering a higher number of networks. Moreover, we will also analyze and study approaches for describing the network flows in terms alternative features.

REFERENCES

[1] W. D. Donato, A. Pescape, and A. Dainotti, "Traffic identification engine: an open platform for traffic classification," *IEEE Network*, vol. 28, no. 2, pp. 56–64, March 2014.

[2] M. Finsterbusch, C. Richter, E. Rocha, J.-A. Muller, and K. Hanssgen, "A survey of payload-based traffic classification approaches," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 1135–1156, 2014.

[3] S. Valenti, D. Rossi, A. Dainotti, A. Pescapè, A. Finamore, and M. Mellia, "Reviewing traffic classification," in *Data Traffic Monitoring and Analysis*. Springer, 2013, pp. 123–147.

[4] T. T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Communications Surveys Tutorials*, vol. 10, no. 4, pp. 56–76, Fourth 2008.

[5] Y. Wang, Y. Xiang, J. Zhang, W. Zhou, and B. Xie, "Internet traffic clustering with side information," *Journal of Computer and System Sciences*, vol. 80, no. 5, pp. 1021 – 1036, 2014.

[6] J. Zhang, X. Chen, Y. Xiang, W. Zhou, and J. Wu, "Robust network traffic classification," *IEEE/ACM Transactions on Networking*, vol. 23, no. 4, pp. 1257–1270, Aug 2015.

[7] T. Bakhshi and B. Ghita, "On internet traffic classification: A two-phased machine learning approach," *Journal of Computer Networks and Communications*, vol. 2016, p. 21, 2016.

[8] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.

[9] J. Cao, Z. Fang, G. Qu, H. Sun, and D. Zhang, "An accurate traffic classification model based on support vector machines," *International Journal of Network Management*, 2017.

[10] M. Fazzolari, R. Alcala, Y. Nojima, H. Ishibuchi, and F. Herrera, "A review of the application of multiobjective evolutionary fuzzy systems: Current status and further directions," *IEEE Transactions on Fuzzy systems*, vol. 21, no. 1, pp. 45–65, 2013.

[11] A. Fernandez, V. Lopez, M. J. del Jesus, and F. Herrera, "Revisiting evolutionary fuzzy systems: Taxonomy, applications, new trends and challenges," *Knowlegde Based Systems*, vol. 80, pp. 109–121, 2015.

[12] J. Camacho, P. Padilla, P. Garca-Teodoro, and J. Daz-Verdejo, "A generalizable dynamic flow pairing method for traffic classification," *Computer Networks*, vol. 57, no. 14, pp. 2718 – 2732, 2013.

[13] M. Antonelli, P. Ducange, and F. Marcelloni, "A fast and efficient multi-objective evolutionary learning scheme for fuzzy rule-based classifiers," *Information Sciences*, vol. 283, pp. 36–54, 2014.

[14] M. Antonelli, P. Ducange, B. Lazzerini, and F. Marcelloni, "Multi-objective evolutionary learning of granularity, membership function parameters and rules of Mamdani fuzzy systems," *Evolutionary Intelligence*, vol. 2, no. 1-2, pp. 21–37, 2009.

[15] A. Segatori, F. Marcelloni, and W. Pedrycz, "On distributed fuzzy decision trees for big data," *IEEE Transactions on Fuzzy Systems*, 2017.

[16] J. Alcala-Fdez, R. Alcala, and F. Herrera, "A fuzzy association rule-based classification model for high-dimensional problems with genetic rule selection and lateral tuning," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 5, pp. 857–872, 2011.

[17] J. R. Quinlan, *C45*. Morgan Kaufmann, 1992.